Nonlocal Segmentation of Point Clouds with Graphs

François Lozes, Moncef Hidane, Abderrahim Elmoataz, Olivier Lézoray

Normandie Univ., UNICAEN, ENSICAEN, GREYC UMR CNRS 6072, Caen, France
{francois.lozes,moncef.hidane,abderrahim.elmoataz,olivier.lezoray}@unicaen.fr

***Abstract*—In this paper, we propose a nonlocal approach based on graphs to segment raw point clouds as a particular class of graph signals. Using the framework of Partial difference Equations (PdEs), we propose a transcription on graphs of recent continuous global active contours along with a minimization algorithm. To apply it on point clouds, we show how to represent a point cloud as a graph weighted with patches. Experiments show the benefits of the approach on raw colored point clouds obtained from real scans[1].**

I. Introduction

There is actually much interest in the development of algorithms that enable to process high-dimensional data that reside on the vertices or edges of a graph, referred to as *graph signals* [1]. With 3D sensors becoming cheaper, more and more applications in natural and applied sciences involve the segmentation of raw 3D colored point clouds. With point clouds, the data to process is not organized on any Cartesian grid and the neighbors of a point have to be defined. Our previous works aim at representing point clouds as weighted graphs to perform nonlocal graph processing [2], [3]. With such a point of view, raw 3D colored point clouds are considered as a specific class of graph signals where a color vector is associated to a point (i.e., a vertex) located in a 3D Euclidean space. In this paper, we propose a new approach for the nonlocal segmentation of point clouds represented by graphs. First, we consider a convex formulation of active contours on graphs, we use the framework of PdEs to obtain a formulation on arbitrary weighted graphs, and propose a minimization strategy. Second, we present a way of associating a patch-based weighted graph to 3D colored point clouds.

II. Weighted Graphs

We recall in this section general definitions and operators relating to graphs. In particular, we review the PdE framework introduced in [4].

A. Notations and Preliminaries

A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists of a finite set $\mathcal{V} = \{v_1, \dots, v_N\}$ of N vertices and a finite set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of weighted edges. We assume $\mathcal{G}$ to be undirected, with no self-loops and no multiple edges. Let (v_i, v_j) be the edge of $\mathcal{E}$ that connects two vertices v_i and v_j of $\mathcal{V}$. Its weight, denoted by $w(v_i, v_j)$, represents the similarity between its vertices. Similarities are usually computed by using a positive symmetric function $w : \mathcal{V}\times\mathcal{V} \to \mathbb{R}^+$ satisfying $w(v_i, v_j) = 0$ if $(v_i, v_j) \notin \mathcal{E}$. The notation $v_i \sim v_j$ is also used to denote two adjacent vertices. The degree of a vertex v_i is defined as $\delta_w(v_i) = \sum_{v_j \sim v_i} w(v_i, v_j)$. Let $\mathcal{H}(\mathcal{V})$ be the Hilbert space of real-valued functions defined on the vertices of a graph. A function $f \in \mathcal{H}(\mathcal{V})$ assigns a real value $f(v_i)$ to each vertex $v_i \in \mathcal{V}$. The $\mathcal{H}(\mathcal{V})$ space is endowed with the usual inner product, denoted $\langle ., . \rangle_{\mathcal{H}(\mathcal{V})}$ in the sequel.

B. Difference Operators on Weighted Graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ be a weighted graph and $w : \mathcal{V}\times\mathcal{V} \to \mathbb{R}^+$ a weight function that depends on the interactions between the vertices. The *difference operator* [4], denoted $d_w : \mathcal{H}(\mathcal{V}) \to \mathcal{H}(\mathcal{E})$, is defined for all $f \in \mathcal{H}(\mathcal{V})$ and $(v_i, v_j) \in \mathcal{E}$ by:

$$(d_w f)(v_i, v_j) = \sqrt{w(v_i, v_j)}(f(v_j) - f(v_i)). \tag{1}$$

The *adjoint* of the difference operator, denoted $d_w^* : \mathcal{H}(\mathcal{E}) \to \mathcal{H}(\mathcal{V})$, is the unique linear operator satisfying $\langle d_w f, H\rangle_{\mathcal{H}(\mathcal{E})} = \langle f, d_w^* H\rangle_{\mathcal{H}(\mathcal{V})}$ for all $f \in \mathcal{H}(\mathcal{V})$ and all $H \in \mathcal{H}(\mathcal{E})$. Its expression is given by:

$$(d_w^* H)(v_i) = \sum_{v_j \sim v_i} \sqrt{w(v_i, v_j)}(H(v_j, v_i) - H(v_i, v_j)). \tag{2}$$

The *divergence operator*, defined by $-d_w^*$, measures the net outflow of a function of $\mathcal{H}(\mathcal{E})$ at each vertex of the graph. The *weighted gradient operator* of a function $f \in \mathcal{H}(\mathcal{V})$, at a vertex $v_i \in \mathcal{V}$, is the vector defined by:

$$(\nabla_w f)(v_i) = ((d_w f)(v_i, v_j))^T_{v_j \in \mathcal{V}}. \tag{3}$$

The ℓ_p norm of this vector is defined, for $p \geq 1$, by:

$$\|(\nabla_w f)(v_i)\|_p = \left(\sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} \big| f(v_j) - f(v_i) \big|^p \right)^{1/p}. \tag{4}$$

III. Segmentation of Point Clouds

In this section, we propose a framework for the segmentation of graph signals (functions defined on the vertices of graphs). To perform the segmentation, we consider a convex formulation of active contours on graphs. Starting from a continuous formulation, we show how to transpose the latter on weighted graphs using the framework of PdEs along with a minimization strategy.

[1]This work was funded under a Ph.D. grant of the regional council of Lower-Normandy.

A. Convex Segmentation on graphs

The usual drawback of active contours methods is the existence of local minimizers and hence their sensitivity to the initial condition. A recent method, introduced by Bresson and Chan in [5], [6], proposes to redefine the active contour model into a model which gives global minimizers. In the continuous setting, where images are viewed as functions on a continuous domain Ω, this model is given by:

$$\underset{f(x)\in\{0,1\}}{\arg\min} \left\{ \int_\Omega ||\nabla f(x)||_1 dx + \lambda \int_\Omega g(f^0)(x) f(x) dx \right\}. \quad (5)$$

The function g is a region detector function used to force region intensity statistics priors. The transposition of (5) on graphs is obtained using the PdEs framework [4], [7] leading to:

$$\bar{f} \in \underset{f:\mathcal{V}\to\{0,1\}}{\operatorname{Arg\,min}} \left\{ \sum_{v_i\in\mathcal{V}} \|(\nabla_w f)(v_i)\|_p^p + \lambda \sum_{v_i\in\mathcal{V}} g(f^0)(v_i) f(v_i) \right\}, \quad (6)$$

where f is a labeling function (± 1 for labeled vertices and 0 for unlabeled ones) and f^0 the signal on the graph (the color vectors associated to the vertices). When $\lambda \neq 0$, this energy can be considered as the nonlocal discrete analogue on graphs of the functional introduced in [6]. We now show how such a minimization can be solved. Problem (6) is non-convex and, as shown in [8] for the continuous analogue, can be reformulated through a convex relaxation. Therefore, a new minimization problem is considered:

$$\hat{f} = \underset{f:\mathcal{V}\to[0,1]}{\arg\min} \left\{ \sum_{v_i\in\mathcal{V}} \|(\nabla_w f)(v_i)\|_p^p + \lambda \sum_{v_i\in\mathcal{V}} g(f^0)(v_i) f(v_i) \right\}. \quad (7)$$

Following the approach in [8], one can show that every level-set of a minimizer of (7) is solution of the original optimization problem (6). As a consequence, to obtain a global solution $\bar{f} : \mathcal{V} \to \{0,1\}$ to the problem of (6), one thresholds any function $\hat{f} : \mathcal{V} \to [0,1]$ that is a solution of (7) and $\bar{f}(v_i) = \chi_\mathcal{S}(v_i)$, where $\mathcal{S} = \{v_i \in \mathcal{V} : \hat{f}(v_i) > t\}$ with $t \in [0,1]$ and χ is the indicator function defined by $\chi : \mathcal{V} \to \{0,1\}$. For a given vertex, if $v_i \in \mathcal{A}$, then $\chi_\mathcal{A}(v_i) = 1$ and $\chi_\mathcal{A}(v_i) = 0$ otherwise. However, to be able to perform such a minimization approach, one has to show that both parts of the energy (7) do verify the co-area formula. This can be easily shown for the second part of the energy (see [8]). We show now that this is also true for the first part.

B. Perimeters and co-area on graphs

Now we show that there is a relation, for the case of a sub-graph, between discrete perimeters on graphs and the co-area formula on graphs.

1) ***Perimeters on graphs:*** Let $\mathcal{A}$ be a set of connected vertices with $\mathcal{A} \subset \mathcal{V}$. We denote by $\partial^+\mathcal{A}$ and $\partial^-\mathcal{A}$, the *external* and *internal* boundary sets of $\mathcal{A}$, respectively. The set $\mathcal{A}^c = \mathcal{V} \setminus \mathcal{A}$ is the complement of $\mathcal{A}$. For a given vertex $v_i \in \mathcal{V}$, one has: $\partial^+\mathcal{A} = \{v_i \in \mathcal{A}^c : \exists v_j \in \mathcal{A} \text{ with } (v_i, v_j) \in \mathcal{E}\}$, $\partial^-\mathcal{A} = \{v_i \in \mathcal{A} : \exists v_j \in \mathcal{A}^c \text{ with } (v_i, v_j) \in \mathcal{E}\}$, and $\partial\mathcal{A} = \{(v_i, v_j) \in \mathcal{E} : \exists v_i \in \partial^+\mathcal{A} \text{ and } v_j \in \partial^-\mathcal{A}\}$. Let us consider non-local regularization functionals based on weighted total variation on graphs $R_{w,p} : \mathcal{H}(V) \to \mathbb{R}$ of a function $f \in \mathcal{H}(V)$: $R_{w,p}(f) = \sum_{v_i\in\mathcal{V}} \|(\nabla_w f)(v_i)\|_p^p$ with $0 < p < +\infty$. By replacing f by the indicator function $\chi_\mathcal{A}$ in these regularization functionals, one has [7]: $R_{w,p}(\chi_\mathcal{A}) = \text{vol}(\partial\mathcal{A}) = Per_{w,p}(\mathcal{A}) = \text{cut}(\mathcal{A}, \mathcal{A}^c)$. This expression can be considered as a nonlocal discrete perimeter of the sub-graph $\mathcal{A}$. Consequently, when $\lambda = 0$ in (7), it is the discrete analogue to the continuous min-cut of [9].

2) ***Co-area formulae on graphs:*** In this subsection, we discuss the co-area formulae on graphs. They are useful on many contexts such as convex relaxation of variational methods on graphs. Let $(\mathcal{V}, \mathcal{E}, w)$ be a weighted graph, $f \in \mathcal{H}(\mathcal{V})$. For $t \in \mathbb{R}$, let $\mathcal{A}_t = \{u \in \mathcal{V} : f(u) > t\}$. Then the co-area formula is verified for $p = 1$ [7] since $Per_{w,1}(\mathcal{A}) = \int_{-\infty}^{\infty} Per_{w,1}(\mathcal{A}_t) dt$. The proof is obvious since $|a - b| = \int_{-\infty}^{+\infty} |\chi_{\{a>t\}} - \chi_{\{b>t\}}| dt$. In the rest of the paper, we will therefore work exclusively with the case of $p = 1$ since $R_{w,1}$ does verify the co-area formulae.

C. Minimization Algorithm on Weighted Graphs

To solve the optimization problem (7), we propose to use the Chambolle and Pock algorithm [10] on weighted graphs, in a similar manner as in [11]. Let us consider the following general optimization problem:

$$\min_{x\in X} F(Kx) + G(x), \quad (8)$$

where X and Y, are two general finite-dimensional vector spaces, and $F \in \Gamma_0(Y)$, $G \in \Gamma_0(X)$ and $K : X \to Y$ a linear operator. The set of all proper convex and lower semicontinuous functions from H to $]-\infty, +\infty]$ is denoted by $\Gamma_0(H)$. Recently, Chambolle and Pock have proposed the following iterative algorithm [10] to solve efficiently (8):

$$\begin{cases} x^0 = \bar{x}^0 = f, \quad y^0 = 0 \\ y^{n+1} = \text{prox}_{\sigma F^*}(y^n + \sigma K \bar{x}^n), \\ x^{n+1} = \text{prox}_{\tau G}(x^n - \tau K^* y^{n+1}), \\ \bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n), \end{cases} \quad (9)$$

where F^* is the conjugate of F [12], K^* is the adjoint operator of K, and prox the proximity operator defined as:

$$\text{prox}_f(x) = \underset{y\in Y}{\arg\min} \left\{ f(y) + \frac{1}{2}||y - x||^2 \right\}. \quad (10)$$

The convergence of algorithm (9) is guaranteed if $\theta = 1$ and $0 < \tau\sigma L^2 < 1$ where $L = ||K|| = \max_{||x||\leq 1} ||Kx||$. The segmentation problem of (7) is formulated with $F = ||.||_1$, $K = \nabla_w$ and $G = \lambda\langle ., g(f^0)\rangle$, with $\langle ., . \rangle$ the dot product operator. By replacing F, K and G, in (9), we can simplify the algorithm. For $y \in Y$, as shown in [11], we have:

$$\text{prox}_{\sigma F^*}(y) = \text{prox}_{\sigma i_B}(y) = \text{proj}_B(y) = \tilde{y},$$

where $\tilde{y}_{ij} = M(y_{ij}) = \frac{y_{ij}}{\max(1, \sqrt{\sum_{v_j \sim v_i} y_{ij}^2})}$ and

$$i_C = \begin{cases} 0 & \text{for } x \in C \\ +\infty & \text{otherwise,} \end{cases} \quad (11)$$

and B is the unitary $||.||_{\infty,2}$ ball.

For $x \in X$, we can show that:

$$\begin{aligned} \text{prox}_{\tau G}(x) &= \arg\min_{y \in Y} \left\{ \tau\lambda\langle y, g(f^0)\rangle + \frac{1}{2}||y - x||^2 \right\} \\ &= x - \tau\lambda g(f^0). \end{aligned} \quad (12)$$

Thus the algorithm to solve the segmentation problem (7) is reduced to:

$$\begin{cases} x^0 = \bar{x}^0 = f, \quad y^0 = 0 \\ y_{ij}^{n+1} = M(y_{ij}^n + \sigma(d_w \bar{x}^n)(v_i, v_j)) \\ x_i^{n+1} = x_i^n - \tau(d_w^* y^{n+1})(v_i) - \tau\lambda g(f^0)(v_i) \\ \bar{x}_i^{n+1} = x_i^{n+1} + \theta(x_i^{n+1} - x_i^n). \end{cases} \quad (13)$$

This algorithm is parametrized by the structure of the graph (topology and weight function w), the functions f, f^0 and $g(f^0)$, and several parameters (λ, τ, θ and σ). On has to note that it is the first time that such a solution is proposed to solve (7) on general weighted graphs.

IV. Construction of a weighted graph from a point cloud

In this section, we explain how a weighted graph based on patches can be associated with a point cloud. This relies on three steps that we detail in the sequel.

A. Graph Creation from Data Points

First step consists in defining the sets $\mathcal{V}$ and $\mathcal{E}$ from a given point cloud. Let us consider a point cloud P as a set of data points $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \in \mathbb{R}^3$. To each data point we first associate a vertex of a proximity graph $\mathcal{G}$ to define a set of vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. Then, determining the edge set $\mathcal{E}$ of the proximity graph $\mathcal{G}$ requires defining the neighbors of each vertex v_i according to its embedding $\mathbf{p}_i$ using the Euclidean distance. We will denote as $\mathcal{D}(v_i, v_j) = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ the Euclidean distance between vertices. We consider the k Nearest Neighbors Graph (k-NNG): $v_j \sim v_i$ if the distance between $\mathbf{p}_i$ and $\mathbf{p}_j$ is among the k-th smallest distances from $\mathbf{p}_i$ to all the other data points. To conclude, the first step consists in associating a k-NNG to the 3D point cloud. The value of k will be denoted k_G for the graph $\mathcal{G}$ associated with the point cloud. To speed up the knn algorithm, a kD-tree is used. Once the graph has been created, it has to be weighted. If one does not want to take care of the vertices similarities, the weight function w can be set to $w = 1$. A better one can be obtained using patches [13]. For images, a patch $\boldsymbol{\mathcal{P}}(v_i)$ centered at a vertex $v_i \in \mathcal{V}$ is a vector of values (e.g., coordinates, intensities, ...) defined by $\boldsymbol{\mathcal{P}}(v_i) = \left(f^0(v_j) : v_j \in B(v_i, n)\right)^T$ where $B(v_i, n)$ is a square of size n^2 centered at v_i. Using patches, $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is defined by $w(v_i, v_j) = exp\left(-\frac{||\boldsymbol{\mathcal{P}}(v_i) - \boldsymbol{\mathcal{P}}(v_j)||_2^2}{\sigma^2}\right)$.

B. Patch Orientation

Patches enable to compute a similarity between two nodes of the graph. It is extensively used with images for inpainting, and restoration. For point clouds, the patch orientation is usually estimated from principal directions computed on a smoothed point cloud [3]. Unfortunately, the obtained orientations of patches are unstable. Because the obtained orientation depends highly on the most predominant axis, one can find different patches orientations for similar points repartitions, and conversely. Recently we have proposed another strategy (see [2]). We have proposed to estimate the patch orientation from the normals. Indeed, this will produce the same orientations for points that have similar normals. The proposed algorithm is therefore to first deduce a tangent vector $\mathbf{t}(v_i)$ from the normal vector $\mathbf{n}(v_i)$. Let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ be the three axis of a 3D space, the tangent vector $\mathbf{t}(v_i)$ is computed with:

$$\begin{cases} \mathbf{t}(v_i) = \mathbf{z} \times \mathbf{n}(v_i) \text{ if } (\mathbf{x} \cdot \mathbf{n}(v_i)) > (\mathbf{z} \cdot \mathbf{n}(v_i)), \\ \mathbf{t}(v_i) = \mathbf{z} \times \mathbf{n}(v_i) \text{ if } (\mathbf{y} \cdot \mathbf{n}(v_i)) > (\mathbf{z} \cdot \mathbf{n}(v_i)), \\ \mathbf{t}(v_i) = \mathbf{x} \times \mathbf{n}(v_i) \text{ otherwise,} \end{cases} \quad (14)$$

with $\times$ the cross product operator, and $\cdot$ the dot product operator. Then a bitangent vector $\mathbf{b}(v_i)$ is computed by $\mathbf{b}(v_i) = \mathbf{n}(v_i) \times \mathbf{t}(v_i)$. The orientations vectors $\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i)$ are then respectively assigned to $\mathbf{t}(v_i), \mathbf{b}(v_i), \mathbf{n}(v_i)$.

C. Patch Construction

Final step consists in constructing the patches. Given a point $\mathbf{p}_i$, defining a patch for this point comes to construct a square grid around $\mathbf{p}_i$ on its tangent plane. We fix the patch length l manually. Let n be the number of cells on a row of the patch. A square lattice of n^2 cells is constructed around $\mathbf{p}_i$ with respect to the basis obtained from orientation computation. Each cell has a side length of l/n. A local graph is then considered that connects the vertex v_i to all the vertices v_j contained in a sphere of diameter $l\sqrt{2}$. Then, all the neighbors v_j of v_i are projected on the tangent plane of $\mathbf{p}_i$ giving rise to projected points $\mathbf{p}'_i$. To fill the patch with values, these projected points $\mathbf{p}'_i$ are associated to the cells the center of which is the closest. The value of the cell is then deduced from a weighted average of the values $f^0(v_j)$ associated with the vertices v_j that where affected to the patch cell. This value is a spectral value (the points' colors). The set of values inside the patch of the vertex v_i are denoted as $\boldsymbol{\mathcal{P}}(v_i)$. Let $C_k(v_i)$ denotes the kth cell of the constructed patch around v_i with $k \in [1, n^2]$. With the proposed patch construction process, one can define the set $V_k(v_i) = \{v_j \mid \mathbf{p}'_j \in C_k(v_i)\}$ as the set of vertices v_j that were affected to the kth patch cell of v_i. Then, the patch vector is defined as $\boldsymbol{\mathcal{P}}(v_i) = \left(\frac{\sum_{v_j \in V_k(v_i)} w(\mathbf{c}_k, \mathbf{p}_j) f^0(v_j)}{|V_k(v_i)|}\right)^T_{k \in [1, n^2]}$, with $w(\mathbf{c}_k, \mathbf{p}_i) = exp(-\frac{||\mathbf{c}_k - \mathbf{p}'_i||_2^2}{\sigma^2})$ and $\mathbf{c_k}$ is the coordinates' vector of the kth patch cell center. This weighting enables to take into account the repartition of the points in the cells' patch to compute their mean feature vectors.

Fig. 1: Segmentation of body parts of a real scanned person (37,161 points) with $k_G = 1000$ and $n^2 = 9$. From left to right respectively: colored raw point cloud, the initial labels, the final labeling.

V. Experiments

This section shows some segmentation results of colored 3D point clouds in two classes using the algorithm presented in (13). The raw data consist in sets of 3D points $\mathbf{p}_i$ (i.e., vertices v_i) that are associated with CIELAB colors vectors ($f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$). In all our experiments, the parameters are $\theta = 1$, $\tau = 1$, $\sigma = 0.25/(w_{max} \times \tau)$ with $w_{max} = \max_{v_i \in \mathcal{V}} \delta_w(v_i)$. The shown results are raw 3D colored point clouds and not meshes, thus some holes can appear in the rendering. Figures 1 and 2 show segmentation results on real and artificial examples with $g(f^0(v_i)) = (\bar{c}_1 - f^0(v_i))^2 - (\bar{c}_2 - f^0(v_i))^2$, where $\bar{c}_1$ and $\bar{c}_2$ are respectively the average colors of both extracted regions. This corresponds to the Chan-Vese model on graphs. The initial partition function f is initialized from seeds provided by the user. Fig. 3 shows a segmentation of the T-shirt of real scanned person. This latter result was obtained by observing that the T-shirt is a heterogenous region, so we used the variance of patches to compute a heterogenous term. We set $g(f^0(v_i)) = (\overline{Var}_1 - f^0(v_i))^2 - (\overline{Var}_2 - f^0(v_i))^2$ where $f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$ represents the variance of $\boldsymbol{\mathcal{P}}(v_i)$ for each color channel, and $\overline{Var}_1, \overline{Var}_2$ are respectively the average variance of patches of extracted regions for each color channel.

VI. Conclusion

In this paper, we have proposed a nonlocal approach based on weighted graphs to segment raw point clouds. We used the framework of PdEs to adapt PDEs on graphs, next we solved the optimization problem of segmentation with the Chambolle and Pock iterative algorithm. We have presented results that show the benefits of the approach on real scanned people and artificial points clouds.

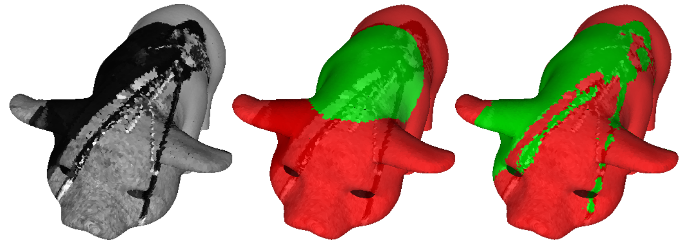

Fig. 2: Segmentation of an artificial point cloud textured with the cameraman image (42,987 points) with $k_G = 1000$ and $n^2 = 9$. From left to right respectively: the cameraman image on a pig point cloud, the initial labels, the final labeling.

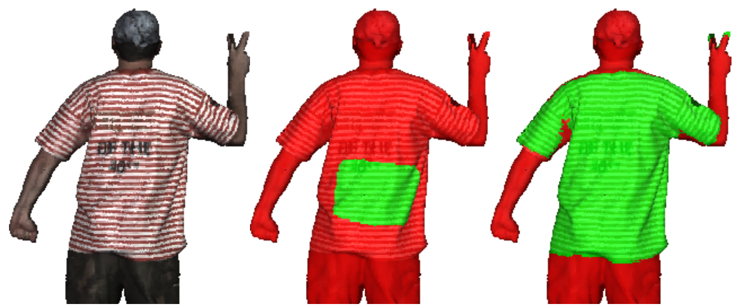

Fig. 3: Segmentation of the heterogenous parts of a real scanned person (76,731 points) with $k_G = 1000$ and $n^2 = 25$. From left to right respectively: colored raw point cloud, person with inital labels, the initial labels, the final labeling.

References

[1] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] F. Lozes, A. Elmoataz, and O. Lézoray, "Morphological PDEs on graphs for filtering and inpainting of point clouds," in *ISPA*, 2013.

[3] F. Lozes, A. Elmoataz, and O. Lézoray, "Nonlocal processing of 3d colored point clouds," in *ICPR*, 2012, pp. 1968–1971.

[4] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE T. Image. Process.*, vol. 17, no. 7, pp. 1047–1060, 2008.

[5] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *J. Math. Imaging Vis.*, vol. 28, no. 2, pp. 151–167, 2007.

[6] X. Bresson and T.F. Chan, "Non-local unsupervised variational image segmentation models," *UCLA CAM Report 08-67*, 2008.

[7] O. Lézoray, A. Elmoataz, and V.-T. Ta, "Nonlocal pdes on graphs for active contours models with applications to image segmentation and data clustering," in *ICASSP*, 2012, pp. 873–876.

[8] T. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models," *SIAM J. Appl. Math.*, vol. 66, no. 5, pp. 1632–1648, 2006.

[9] N. Houhou, X. Bresson, A. Szlam, T. F. Chan, and J.-P. Thiran, "Semi-supervised segmentation based on non-local continuous min-cut," in *SSVM*, 2009, vol. LNCS 5567, pp. 112–123.

[10] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011.

[11] M. Hidane, O. Lézoray, and A. Elmoataz, "Nonlinear Multilayered Representation of Graph-Signals," *J. Math. Imaging Vis.*, vol. 45, no. 2, pp. 114–137, 2013.

[12] H.H. Bauschke and P.L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, 2011.

[13] A.Buades, B.Coll, and J.-M. Morel, "Image denoising methods. a new nonlocal principle," *SIAM Review*, vol. 52, no. 1, pp. 113–147, 2010.